



TITLE:

Simple Loop ProgramによるTime Complexityの階層 (計算機科学の数学的基礎)

AUTHOR(S):

笠井, 琢美; 足立, 暁生

CITATION:

笠井, 琢美 ...[et al]. Simple Loop ProgramによるTime Complexityの階層 (計算機科学の数学的基礎). 数理解析研究所講究録 1978, 322: 20-55

ISSUE DATE:

1978-03

URL:

<http://hdl.handle.net/2433/104033>

RIGHT:

Simple Loop Program による

Time complexity の階層

京大 数理解

笠井 琢美

日本 IBM

尾立 暁生

§ 1 序

従来の loop program による階層は、実際の計算可能性という見地からみて広すぎる。loop のネストの深さが 2 の loop program によって実現される関数のクラス \mathcal{L}_2 が初等関数のクラスと一致してしまう。ここでは loop program を制限することによって time complexity のもっと細かい階層を研究する。即ち simple loop program という loop program の制限されたクラスを導入する。これによって 2 種類の階層 $\mathcal{L}_0 \subset \mathcal{L}_1 \subset \dots$ と $\mathcal{E}_{LP}^0 \subset \mathcal{E}_{LP}^1 \subset \dots$ を syntactical に characterize する。ここで \mathcal{L}_k は $O(n^k)$ 時間で認識できる言語のクラス、 \mathcal{E}_{LP}^k は n の長さの指数関数時間で認識できる言語のクラスである。ここで n は入力の長さである。しに $\bigcup_{k=0}^{\infty} \mathcal{L}_k = \mathcal{E}_{LP}^0$ は多項式時間認識可能な言語のクラス、 $\bigcup_{k=0}^{\infty} \mathcal{E}_{LP}^k$ は初等的問題のク

ラスである。

一般にプログラムの time complexity は計算可能でない。
しかし、ここで述べるように syntax を制限すれば計算量のよい上界が、そのプログラムを syntactical に解析することによって得ることが出来る。

§2. loop program on SM.

Def. X, X', X'' を互いに素な可算集合とする。 X の元を control variable, X' の元を variable, X'' の元を array name と呼ぶ。 loop program とは次で定義される statement のことをいう。 ここで i, j は variable, x は control variable, u は variable か又は control variable, A は array name, c は $N = \{0, 1, \dots\}$ の元を表わす。

$\langle \text{atomic statement} \rangle ::= u \leftarrow u+1 \mid u \leftarrow u-1$

$\mid u \leftarrow c \mid A[u] \leftarrow v \mid c \leftarrow A[u]$

$\langle \text{loop statement} \rangle ::= \text{loop } x \text{ do } \langle \text{statement} \rangle \text{ end}$

$\langle \text{condition} \rangle ::= u = c \mid u \neq c$

$\langle \text{statement} \rangle ::= \langle \text{atomic statement} \rangle \mid \langle \text{loop statement} \rangle$

$\mid \text{if } \langle \text{condition} \rangle \text{ then } \langle \text{statement} \rangle \text{ else } \langle \text{statement} \rangle \text{ end}$

$\mid \langle \text{statement} \rangle ; \langle \text{statement} \rangle$

Def. ステップ機 (SM と略す) のプログラムとは列

$$\alpha_1; \alpha_2; \dots; \alpha_k$$

のことをいう。ここで α_i は次のいずれか

(i) do <atomic statement> then goto l

(ii) if <condition> then goto l else goto l'

(iii) halt,

但し $1 \leq l, l' \leq k$.

Def. \tilde{X} で変数および添字付変数の集合を表わす。すな

わら、 $\tilde{X} = X \cup X' \cup \{A[i]: A \in X, i \in \mathbb{N}\}$. 関数

$d: \tilde{X} \rightarrow \mathbb{N}$ を memory map と呼ぶ。memory map 全体

からなる集合を D と書く。 P を loop program (SM のプ

ログラム) とする。すると、 P は D から D への関数 (部

分関数) を実現する。 P の実現する関数を同じ記号

$P: D \rightarrow D$ で表わす。 P の time complexity とは関数

$\text{time}_P: D \rightarrow \mathbb{N}$ で、 $\text{time}_P(d)$ は初期 memory d のもとで P

が実行する atomic statement の個数を表わす。

Remark. loop program は SM のプログラムの制限さ

れたものとみることが出来る。ここでの loop program は

本来のもの [3, 4, 9] とは多少異なる。従来の loop program

では $u \leftarrow u + 1$ の形のステートメント, 及び if-then-else を含んでいないし, 配列も許していない。

$u \leftarrow u + 1$ と if-then-else を許した理由は, これらをなして表現するためには, loop が 1 個余分に必要とするからであり, 配列を許した理由は, 配列を含むプログラムを配列を含まないプログラムで “多項式時間” では simulate できないからである [7]。

Def. P を loop program とする。このとき X 上の関係 \succsim_P を次のように定める。 $x \succsim_P y$ であるとは, P が loop x do Q end の形の statement を含み, Q が $y \leftarrow y + 1$ を含むときをいう。 P が simple であるとは,

$$x_1 \succsim_P x_2 \succsim_P \dots \succsim_P x_k, \quad k > 1, \quad x_1 = x_k$$

なる列を含まないときをいう。したがって P が simple ならば \succsim_P の reflexive transitive closure は X 上の partial order である。

Def. P を loop program, A を P に現われる atomic statement の occurrence とする。このとき A の 深さ とは A を含む loop statement の個数をいう。 A_1, \dots, A_k を P の深さ ≥ 1 の atomic statement の occurrence とする。

t_i を A_i の深さとする。このとき t_1, t_2, \dots, t_k を P の loop complexity と呼び、 $l_c(P)$ で示す。

例 1.1. P : loop x do
 loop y do $z \leftarrow z+1$ end
 loop x do
 loop x do $y \leftarrow y+1$ end end end

とする。すると $x \geq_P y$, $y \geq_P z$. よって P は simple であり、 $l_c(P) = 2 \times 3 = 6$ である。

以下で loop complexity によって時間量の上限が計算できることを示す。また多項式時間計算可能な言語のクラスと loop complexity との関係について調べる。§5 ではプログラム P の時間量のもと正確な解析をする方法について述べる。

§3 基本的性質

Def. プログラム P が reduced であるとは、 P が $x \leftarrow x+1$, x は control variable, の形の statement と loop statement だけから構成されることをいう。 P を program とするとき、 P の reduced form \bar{P} を次のように定める。

(2"んた37°ロ7"54にえ)

$Z \in$ 現われ"ない control variable とする。 P の

$x \leftarrow x+1$ の形以外の atomic statement なら \bar{P} は $Z \leftarrow Z+1$,

P の if q then P_1 else P_2 なら \bar{P} は $\bar{P}_1; \bar{P}_2$, P の loop

x do P_1 end なら \bar{P} は loop x do \bar{P}_1 end.

Def. 各 $k \in \mathbb{N}$ に対し, $S_k: D \rightarrow D$ を次のように定める。 各 $x \in \tilde{X}$ に対し

$$S_k(d)(x) = \begin{cases} d(x) & \text{if } d(x) \geq k \\ n & \text{otherwise} \end{cases}$$

$d, d' \in D$ の元とする。 任意の $x \in X$ に対し, $d(x) \leq d'(x)$ となるとき $d \leq d'$ と書く。

Lemma 3.1 Q を reduced な loop program とする。

- (i) 任意の $d \in D$ に対し, $d \leq Q(d)$
- (ii) $d \leq d'$ なら $Q(d) \leq Q(d')$
- (iii) $d \leq d'$ なら $\text{time}_Q(d) \leq \text{time}_Q(d')$

Proof. 明らか。

Lemma 3.2 P を loop program, \bar{P} を P の reduced form とすると (i) $\ell_c(P) = \ell_c(\bar{P})$, (ii) $k \in \mathbb{N}$ が存

とし任意の $d \in D$ に対し

$$\text{time } p(d) \leq \text{time } \bar{p}(S_k(d))$$

Proof. (i) は \bar{p} の構成より明らか。 \bar{p} の構成による帰納法で, (ii) と次の (3.1) を示す。

(3.1) $k \in \mathbb{N}$ が存在し、任意の $d \in D$ に対し

$$p(d) \leq \bar{p}(S_k(d))$$

が成立する。 p が atomic statement なら (ii) 及び (3.1) は明らかに成立。 (p が $x \leftarrow c$, $c \in \mathbb{N}$, のとき, $k = c$ とすると, (3.1) が成立する)

p が if q then P_1 else P_2 end のとき, 帰納法の仮定より, k_1, k_2 が存在し

$$\text{time } p(d) \leq \text{time } \bar{p}(S_k(d))$$

$$P_i(d) \leq \bar{P}_i(S_{k_i}(d))$$

が成立する。 $k = \max\{k_1, k_2\}$, $d' = \bar{P}_1(S_k(d))$ とすると Lemma 3.1 より

$$S_k(d) \leq d'$$

より

$$\begin{aligned} \text{time } p(d) &= \max\{\text{time } p_1(d), \text{time } p_2(d)\} \\ &\leq \text{time } p_1(d) + \text{time } p_2(d) \\ &\leq \text{time } \bar{p}_1(S_k(d)) + \text{time } \bar{p}_2(S_k(d)) \end{aligned}$$

$$\begin{aligned} &\leq \text{time } \bar{p}_1(S_k(d)) + \text{time } \bar{p}_2(d') \\ &= \text{time } \bar{p}_1, \bar{p}_2(S_k(d)) = \text{time } \bar{p}(S_k(d)) \end{aligned}$$

よって (iii) が成立する。 $d \in D$, $x \in X$ に対し

$$\begin{aligned} P(d)(x) &= \max \{ P_1(d)(x), P_2(d)(x) \} \\ &\leq \max \{ \bar{P}_1(S_k(d))(x), \bar{P}_2(S_k(d))(x) \} \\ &\leq \bar{P}_2(\bar{P}_1(S_k(d))(x)) = \bar{P}(S_k(d))(x) \end{aligned}$$

よって (3.1) が成立。

P が `loop x do P_1 end` のとき, $d_0 \in D$, $d_0(x) = t$ とし,

$$d_{i+1} = P_1(d_i),$$

$$d'_0 = S_k(d_0), \quad d'_{i+1} = \bar{P}_1(d_i), \quad 0 \leq i < t$$

と置く。帰納法の仮定より 長が存在し

$$P_1(d_i) \leq \bar{P}_1(S_k(d_i))$$

$$\text{time } p_1(d_i) \leq \text{time } \bar{p}_1(S_k(d_i))$$

が成立する。しに関する帰納法でまず

$$(3.2) \quad S_k(d_i) \leq d'_i$$

を示す。 $i=0$ のときは明らか。 i で成立するとする。

$$S_k(d_{i+1}) = S_k(P_1(d_i))$$

$$\leq S_k(\bar{P}_1(S_k(d_i)))$$

$$= \bar{P}_1(S_k(d_i)) \leq \bar{P}_1(d'_i) = d'_{i+1}.$$

よって (3.2) はすべての i , $0 \leq i < t$ で成立する。よって

$$\text{time } p(d_0) = \sum_{i=0}^{t-1} \text{time } p_1(d_i)$$

$$\begin{aligned}
 &\leq \sum_{i=0}^{k-1} \text{time } \bar{p} (S_k(d_i)) \\
 &\leq \sum_{i=0}^{k-1} \text{time } \bar{p}_1 (d_i) = \text{time } \bar{p} (d_0)
 \end{aligned}$$

よって (ii) が成立。 また

$$P(d_0) = dt \leq S_k(dt) \leq d'_k = \bar{p}(d_0)$$

より (3.1) が成立。 $P = P_1; P_2$ のときも同様である。

Def. P と Q を reduced な loop program とする。このとき任意の $d \in D$ に対し $P(d) \leq Q(d)$ となるとき $P \leq Q$ と書く。 P を reduced なら

$$\text{time}_p(d) = \sum_{x \in x} (P(d)(x) - d(x))$$

である。 よって次が成立する。

Lemma 3.3 P, Q を reduced とする。 $P \leq Q$ なら任意 $d \in D$ に対し

$$\text{time}_p(d) \leq \text{time}_q(d)$$

Lemma 3.4 P, Q, Q' を reduced とし、 $Q \leq Q'$ とする。 P が Q を含むとき、 P の Q を Q' で置き替えて得られる program を P' とする。 すると

$$P \leq P'.$$

Def. P を loop program とする。 $x \in X$ とする。

P が $x \leftarrow x + 1$ を含むなら x は P の right variable とい
い、 P が $\text{loop } x \text{ do } Q \text{ end}$ の形の statement を含むとき
 x は P の control variable といふ。 P の right variable
全体からなる集合を $R_v(P)$, Control variable 全体からな
る集合 $C_v(P)$ と書く。

Lemma 3.5 P, Q を reduced な loop program とする。

$$R_v(P) \cap C_v(Q) = \emptyset \text{ なら}$$

$$P \circ Q \leq Q \circ P$$

Proof. R を reduced な loop program とする。:

のとき R の構成に因する帰納法で次を示すことが出来る。

(3.3) 任意の $x \in C_v(R)$ に対し $d(x) \leq d'(x)$ なら
任意の $y \in X$ に対し

$$R(d)(y) - d(y) \leq R(d')(y) - d'(y).$$

$d_0 \in D$, $d_1 = P(d_0)$, $d_2 = Q(d_1)$, $d'_1 = Q(d_0)$, $d'_2 = P(d'_1)$
と置く, すると任意の $x \in C_v(Q)$ に対し $d_0(x) = d_1(x)$ となるから (3.3) より任意の $y \in X$ に対し

$$(3.4) \quad d_2(y) - d_1(y) = d'_1(y) - d_0(y)$$

が成立する。 また任意の $x \in C_v(P)$ に対し $d_0(x) \leq d'_1(x)$

が成立するから, (3.3) より任意の $y \in X$ に対し

$$(3.5) \quad d_1(y) - d_0(y) \leq d'_2(y) - d'_1(y)$$

が成立する, (3.4) と (3.5) より

$$d_2(y) \leq d'_2(y) \quad \text{for all } y \in X$$

となり $d_2 \leq d'_2$ となる。

Def. loop program の normal form を次のように定義する。ここに x は control variable を表す。

$$\langle \text{type 1} \rangle ::= x \leftarrow x + 1$$

$$| \text{loop } x \text{ do } \langle \text{type 1} \rangle \text{ end}$$

$$\langle \text{normal form} \rangle ::= \langle \text{type 1} \rangle \mid \langle \text{normal form} \rangle :$$

$$\langle \text{normal form} \rangle$$

Theorem 3.6 P を reduced if simple loop program とする。このとき normal form の simple loop program Q で次を満たすものが存在する。

$$(i) \quad l_c(P) = l_c(Q) \quad (ii) \quad P \leq Q$$

Proof. P が次の形の statement を含んでいるとする

$$\text{loop } x \text{ do } P_1; P_2 \dots, P_k \text{ end}, \quad k > 1$$

とする。

$$(3.7) \quad Cr(P_i) \cap Rv(P_j) = \emptyset \text{ for all } j, 1 \leq j \leq k$$

と仮定してよいことを示す。 P_1 は simple であるから、

あるしか存在し $Cr(P_1)$ なる任意の j ($1 \leq j \leq k$) に対し

$$Cr(P_1) \cap Rv(P_j) = \emptyset$$

が成立する。 Lemma 3.4 と 3.5 より

$$P_1; P_2; \dots; P_k \leq P_1; P_1; \dots; P_{i-1}; P_{i+1}; \dots; P_k.$$

よって左辺を右辺に置きかえるにものを新たに P とすればよい。

(3.7) が成立したとする。 このとき

$$(3.8) \quad \begin{aligned} \text{loop } x \text{ do } P_1; \dots; P_k \text{ end} \\ \leq \text{loop } x \text{ do } P_1 \text{ end; loop } x \text{ do } P_2; \dots; P_k \text{ end.} \end{aligned}$$

が成立すること示す。 $d \in D$, $d(x) = t$ とする。

すると Lemma 3.5 より

$$\begin{aligned} \text{loop } x \text{ do } P_1; \dots; P_k \text{ end} \\ = (P_1; \dots; P_k)^t \leq P_1^t; (P_2; \dots; P_k)^t \\ = \text{loop } x \text{ do } P_1 \text{ end; loop } x \text{ do } P_2; \dots; P_k \text{ end.} \end{aligned}$$

が成立する。上述の操作をくり返すことによって目的の

Q が得られる。

Theorem 3.7 P is simple loop program, $d \in D$,

$m = \max_{x \in X} d(x)$ とおく。すると定数 C が存在し

$$\text{time}_P(d) \leq C m^{l_c(P)}$$

Proof P の reduced form を \bar{P} とする。 P が simple ならば \bar{P} も simple. Lemma 3.2 より $l_c(P) = l_c(\bar{P})$ である。

$$(3.8) \quad \text{time}_P(d) \leq \text{time}_{\bar{P}}(S_k(d)) \text{ for some } k.$$

Theorem 3.6 と Lemma 3.3 より, normal form の simple loop program Q が存在し $l_c(\bar{P}) = l_c(Q)$ である。

$$(3.9) \quad \text{time}_{\bar{P}}(d) \leq \text{time}_Q(d)$$

が成立する。よって任意の normal form の simple loop program Q に対し

$$(3.10) \quad \text{time}_Q(d) \leq C (\max_x d(x))^{l_c(Q)} \text{ for some } C.$$

が示される。以下に (3.8), (3.9), (3.10) より

$$\begin{aligned} \text{time}_P(d) &\leq \text{time}_{\bar{P}}(S_k(d)) \\ &\leq \text{time}_Q(S_k(d)) \\ &\leq C (\max_x d(x) + k)^{l_c(Q)} \\ &\leq C' (\max_x d(x))^{l_c(P)} \end{aligned}$$

が成立するからである。(3.10) を Q の構成に因する帰納法で示す。 Q が type 1 ならば明らかに成立する。

$Q = Q_1; Q_2$, $l_c(Q_i) \geq 1$, とする。 $d' = Q_1(d)$ と

らく。すると

$$\begin{aligned} \max_x d'(x) &\leq m + \text{time}_{Q_1}(d) \\ &\leq m + c_1 m^{l_1(Q_1)} \end{aligned}$$

よって

$$\begin{aligned} \text{time}_Q(d) &= \text{time}_{Q_1}(d) + \text{time}_{Q_2}(d') \\ &\leq c_1 m^{l_1(Q_1)} + c_2 (m + c_1 m^{l_1(Q_1)})^{l_2(Q_2)} \\ &\leq c_3 m^{l_1(Q_1) \cdot l_2(Q_2)} \quad \text{for some } c_3 \end{aligned}$$

となり (3.10) が成立する。

§ 4. Polynomial time computability

多くの問題は言語の認識問題として形式化される。

$[k] = \{1, 2, \dots, k-1\}$ を alphabet, $[k]^*$ の subset を 言語 と呼ぶ。言語 $L \subset [k]^*$ が SM のプログラム P で $f(n)$ 時間認識可能であるとは, P に $na_1 a_2 \dots a_n$, $n \in \mathbb{N}$, $a_i \in [k]$, が入力として与えられたとき $f(n)$ 時間以内に P が停止し, $a_1 \dots a_n \in L$ なら 1 をそうでないなら 0 を出力するときをいう。(たとえば, 変数 x に入力の長さ n が, 配列 $A[1], \dots, A[n]$ に a_1, \dots, a_n が set された状態で P は計算を開始し, 停止したときの変数 y の値が 1 なら受理, 0 なら拒否とする。) L が多テープ Turing Machine (TM と略す) で $f(n)$ 時間認識可

能ならず, L は SM で $O(f(n))$ 時間認識可能ならず, L は ITM で $O(f^2(n))$ 時間認識可能である。

Notation 各 $k \in \mathbb{N}$ に対し, P_k で SM で $O(n^k)$ 時間認識可能な言語のクラスを表わす。 $P = \bigcup_{k=0}^{\infty} P_k$ とおく。しにがって P は多項式時間認識可能な言語のクラスを表わす。次にプログラムのクラスを simple loop program に制限しても, 時間に関しては, それほど不経済とはならないことを示す。証明は本質的に Constable and Bordin に負う。

Lemma 4.1 L は SM のプログラム P で Cn^k 時間認識可能であつたとする。すると L は loop complexity が k である simple loop program Q で受理できる。

Proof simple loop program $P_i, 1 \leq i \leq k$, を次のように定める。

$$P_0 = y \leftarrow y + 1$$

$$P_{i+1} = \text{loop } x \text{ do } P_i \text{ end}$$

また R を次のように定める。

$$y \leftarrow 0; \quad z \leftarrow 0; \quad P_k; \quad (\text{loop } y \text{ do } \overset{(z \leftarrow)}{z+1} \text{ end})^{(c)}$$

$$d' = (y \leftarrow 0; \quad z \leftarrow 0; \quad P_k)(d), \quad d'' = R(d), \quad d(x) = n$$

とすると $d'(y) = n^k$, $d'(z) = \lceil c \rceil, n^k$ とする。 P は次の3つの型の statement の有限集合である

(4.1) $i: \text{do } \alpha \text{ then goto } l$

(4.2) $i: \text{if } q \text{ then goto } l \text{ else goto } l'$

(4.3) $i: \text{halt}$

ここで α は atomic statement, q は判定である。また P は control variable を含まないとしてよい。 Q は次のように定められる。

$R; w \leftarrow 0; \text{loop } z \text{ do } R_0 \text{ end}$

ここで 各 R_i は次のように定められる。 P の i 番目の statement が (4.1) の形なら

$R_i = \text{if } w=0 \text{ then } i: w \leftarrow l \text{ else } w \leftarrow w-1; R_{i+1} \text{ end.}$

P の i 番目の statement が (4.2) の形なら

$R_i = \text{if } w=0 \text{ then}$
 $\text{if } q \text{ then } w \leftarrow l \text{ else } w \leftarrow l' \text{ end}$
 $\text{else } w \leftarrow w-1, R_{i+1}$

(4.3) の形なら

$R_i = \text{if } w=0 \text{ then } w \leftarrow i \text{ else } w \leftarrow w-1; R_{i+1} \text{ end.}$

このように構成された Q は明らかに P と同値である。

よに $l_c(Q) = l_c(P_k) = k$.

Lemma 4.1 と Theorem 3.7 より次の定理が得られる。
この定理は P_k の一つのプログラムによる syntactic characterization を与える。

Theorem 4.2 $L \in P_k$ である必要十分条件は loop complexity k の simple loop program で L が受理されることである。

Corollary 4.3 $L \in P$ である必要十分条件は L が simple loop program で受理されること。

§5 精密な解析

loop complexity により simple loop program の時間量のららさつは上界が得られる。たとえば例 11 のプログラム P を考える。 $d \in D$, $n = \max_x d(x)$ とすると

P の loop complexity は 6 から

$$\text{time}_P(d) \leq cn^6$$

となる。しかし、もっと正確には

$$\text{time}_P(d) \leq cn^4$$

がいえ。ここでは与えられた simple loop program の
時間量の上限をもっと正確に求める方法について述べる。

Def P を解析しようとするプログラムとする。 X_P で
 P に現われる control variable の集合とする。このと
き P の term を次のように定義する。ここで c は N の
元 x は X_P の元を表わす。

$$\langle \text{term} \rangle ::= c \mid x \mid (\langle \text{term} \rangle \cdot \langle \text{term} \rangle)$$

$$\mid (\langle \text{term} \rangle + \langle \text{term} \rangle) \mid \max \{ \langle \text{term} \rangle, \dots, \langle \text{term} \rangle \}$$

$d \in D$, $t \in \text{term}$ とする。このとき $d(t) \in N$ を次のよ
うに定義する。

$$d(c) = c, \quad c \in N$$

$$d(t_1 + t_2) = d(t_1) + d(t_2)$$

$$d(t_1 \cdot t_2) = d(t_1) \cdot d(t_2)$$

$$d(\max \{ t_1, \dots, t_m \}) = \max \{ d(t_1), \dots, d(t_m) \}$$

$X_P = \{ x_1, \dots, x_n \}$ とする。ここで P の各 substatement
 Q に対し、以下の性質 (5.1), (5.2) を持つような term の
列 $V(Q), U(Q)$ を求める algorithm について考える。

(5.1) $V(Q)$ は term の n 組 (t_1, \dots, t_n) で任

意の $d \in D$, $x_i \in X_P$ に対し

$$Q(d)(x_i) \leq d(t_i)$$

が成立する。

(5.2) Q が 1 個の atomic statement を持つ
 なら, Q は term の長組 (u_1, \dots, u_k) で, 任意の $d \in D$ に対し,
 d のもとで Q を実行したとき, Q の i 番目の atomic
 statement が実行される回数 $d(u_i)$ 回以内である。さ
 らに次の algorithm で構成される $V(Q)$ は次の性質を持つ。

(5.3) $V(Q)$ の i 番目の component v_i は次の形。

$$\max \{v_1, \dots, v_m\}$$

ここで 各 v_j , $1 \leq j \leq m$, は変数 x_l を含まないか, $v_j = x_l$
 があるいは $v_j = x_l + v'_j$ の形で, v'_j は x_l を含まない。

さらに v_j が x_l ($l \neq i$) を含めば $x_l <_p x_i$ である。特
 に $m=1$ のとき, $\max \{v_i\}$ のかわりに v_i と書く。

Algorithm 5.1

Input: simple loop program P

Output: (5.2) を満たす $V(P)$

方法: P の各 substatement Q に対し, (5.1)~(5.3)
 を満たす $V(Q)$, $U(Q)$ を順に構成していく。 $X_P = \{x_1, \dots, x_n\}$
 とする。

Case 1 Q の atomic statement のとき

$$U(Q) = \{1\}$$

$V(Q) = (t_1, \dots, t_n)$ は次のように定められる。

Case 1.1 Q が $x_i \leftarrow x_{i+1}$ のとき

$$t_j = \begin{cases} x_j & \text{if } j \neq i \\ x_{i+1} & \text{if } j = i \end{cases}$$

Case 1.2 Q が $x_i \leftarrow c$, $c \in N$ のとき

$$t_j = \begin{cases} x_j & \text{if } j \neq i \\ c & \text{if } j = i \end{cases}$$

Case 1.3 Q が 上の atomic statement のとき

$$t_j = x_j \quad 1 \leq j \leq n$$

Case 2 $Q = \text{if } q \text{ then } Q_1 \text{ else } Q_2 \text{ end}$ のとき

$$U(Q) = U(Q_1) \cup U(Q_2)$$

と定める。 $V(Q_1) = (\max u_1, \max u_2, \dots, \max u_n)$, $V(Q_2)$

$= (\max v_1, \dots, \max v_n)$ とする。 このとき $V(Q) \in$

$$V(Q) = (\max u_1 \cup v_1, \max u_2 \cup v_2, \dots, \max u_n \cup v_n)$$

と定める。

Case 3. $Q = Q_1; Q_2$ のとき. $V(Q_1) = (u_1, \dots, u_n)$
 $V(Q_2) = (v_1, \dots, v_n)$ とする. t を任意の Term とす
 るとき, \tilde{t} と t に現われる x_i を u_i で置き替えて得られ
 る Term を表わす. $U(Q)$ は次で定められる.

$$U(Q) = U(Q_1) \cup \{ \tilde{t} : t \in U(Q_2) \}$$

$V(Q) = (t_1, \dots, t_n)$ は次のように定義される. $u_i =$
 $\max \{ u_{i1}, \dots, u_{ij_i} \}$, $v_i = \max \{ v_{i1}, \dots, v_{i k_i} \}$ とする.
 $t_i = \max (\{ \tilde{v}_{if} : v_{if} \text{ の } x_i \text{ を含まない, } 1 \leq f \leq k_i \}$
 $\cup \{ u_{ig} : 1 \leq g \leq j_i, \text{ if } v_{if} = x_i \text{ for some } f \}$
 $\cup \{ u_{ig} + \tilde{w}_f : 1 \leq g \leq j_i, v_{if} = x_i + w_f \text{ の形のもの} \})$

Case 4. $Q = \text{loop } x_i \text{ do } Q_1 \text{ end}$ のとき. $V(Q_1)$
 $= (u_1, u_2, \dots, u_n)$ より $V(Q) = (t_1, \dots, t_n)$ を求める.
 ここで $<_p$ の大きい順に t_i を計算する. したがって
 $x_i <_p x_j$ なら t_j を t_i より先に計算する.

今, $x_i <_p x_j$ ならすべての j に対し, t_j が求まると
 する. $u_i = \max \{ v_{i1}, \dots, v_{if}, x_i + w_1, x_i + w_2, \dots, x_i + w_{f_i} \}$
 又は $u_i = \max \{ v_{i1}, \dots, v_{if}, x_i + w_1, \dots, x_i + w_{f_i}, x_i \}$ と
 する. ことにし $f \geq 0, g \geq 0$ で v_{if}, w_{ig} は x_i を
 含まない term.

このとき t_i を

$$t_i = \max \{ \max \{ \tilde{v}_1, \dots, \tilde{v}_f \} + x_e \cdot \max \{ \tilde{w}_1, \dots, \tilde{w}_g \}, \\ x_i + x_e \cdot \max \{ \tilde{w}_1, \dots, \tilde{w}_g \} \}$$

と定める。ここで \tilde{w} は term より, w に現われる各変数 x_f を t_f で置き替えて得られる term. (v_e 又は w_e が変数 x_f を含めば (5.3) より $x_i <_P x_f$. よって t_f はすでに求まっている.) $U(Q)$ は次で定められる.

$$U(Q) = \{ x_e \cdot \tilde{t} : t \in U(Q_i) \}$$

Algorithm 5.2.

Input : simple loop program P

Output : term t such that, for all $d \in D$,
 $\text{time}_P(d) \leq t(d)$

方法 : $U(P)$ を Algorithm 5.1 で計算し

$$t = \sum_{t' \in U(P)} t' \quad \text{とする.}$$

例 5.3 次のプログラム P を考える

$y \leftarrow 0; z \leftarrow 0; w \leftarrow 0;$

loop x do

if $A(y) = 0$ then

$w \leftarrow 3;$

loop y do

$z \leftarrow z + 1; w \leftarrow w + 1$ end

else

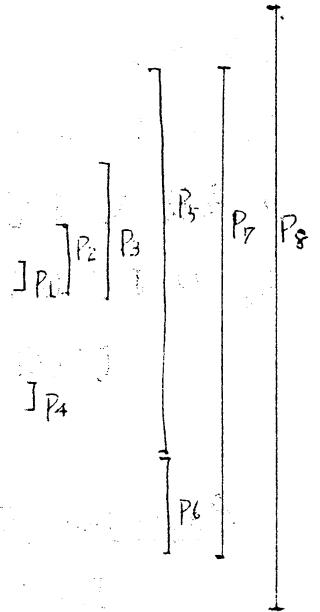
loop z do $w \leftarrow w + 1$ end

end;

loop w do $B(w) \leftarrow w$ end;

$y \leftarrow y + 1$

end



P は simple である。次に Algorithm 5.1 に従って $V(P_1)$, $U(P_1)$ を計算する。 $V(P_i)$ の変数 x に対応する要素が t なら $x \leftarrow t$ と書く。また $x \leftarrow x$ の形は省略する。

$$V(P_1) = \{z \leftarrow z + 1, w \leftarrow w + 1\}, U(P_1) = \{1, 1\}$$

$$V(P_2) = \{z \leftarrow z + y, w \leftarrow w + y\}, U(P_2) = \{y, y\}$$

$$V(P_3) = \{z \leftarrow z + y, w \leftarrow 3 + y\}, U(P_3) = \{1, y, y\}$$

$$V(P_4) = \{w \leftarrow w + z\}, U(P_4) = \{z\}$$

$$V(P_5) = \{w \leftarrow \max\{3 + y, w + z\}, z \leftarrow z + y\},$$

$$U(P_5) = \{1, y, y, z\}$$

$$V(P_6) = \{y \leftarrow y + 1\},$$

$$U(P_6) = \{w, 1\}$$

$$V(P_7) = \{w \leftarrow \max\{3+y, w+z, w+x\}, z \leftarrow z+y,$$

$$y \leftarrow y+1\} \quad U(P_7) = \{1, y, y, z, w, 1\}$$

$$V(P_8) = \{y \leftarrow y+x, z \leftarrow z+x(y+x),$$

$$w \leftarrow \max\{3+y+x, w\} + x(z+x(y+x))\}$$

($w < p < z < p < y$ であるから, y, z, w , の順に計算する)

$$U(P_8) = \{x, x(y+x), x(y+x), x(z+x(y+x)),$$

$$x(\max\{3+y+x, w\} + x(z+x(y+x))), x\}$$

よって求める $U(P)$ は次のようになる

$$U(P) = \{x, x^2, x^2, x^2, x(3+x+x^2), x, 1, 1, 1\}$$

よって

$$\text{time}_P(d) \leq d(x^4 + x^3 + 3x^2 + 5x + 3)$$

Theorem 5.4 Algorithm 5.1 は $U(P)$ を正しく計算する。

る。

Proof Q の構成に関する帰納法で, $U(Q)$, $V(Q)$ が (5.1) ~ (5.3) を満たすことを証明する。Case 1, Case 2 の場合は明らかである。

Case 3 の場合を考える。構成より $V(Q)$ が (5.3) を満たすことは明らか。(ただし $x_i, j \neq i$ を含み u_j が x_i を含むことはない。もしそうだとすると, $x_i < p < x_j$,

$x_i <_P x_i$ となり, P が simple であることに矛盾する.)

$v_{if} = x_i + w$ とする $1 \leq f \leq k_i$. すると

$$\begin{aligned}\tilde{v}_{if} &= \max \{ u_{i1}, \dots, u_{if}, \dots, u_{ik_i} \} + \tilde{w} \\ &= \max \{ u_{i1} + \tilde{w}, \dots, u_{if} + \tilde{w}, \dots, u_{ik_i} + \tilde{w} \}\end{aligned}$$

よって

$$t_i = \max \{ \tilde{v}_{i1}, \dots, \tilde{v}_{ik_i} \} = \tilde{v}_i$$

となる. $d' = Q_1(d)$, $d'' = Q_2(d)$ とする.

帰納法の仮定より

$$d'(x_i) \leq d(u_f), \quad 1 \leq f \leq n$$

$$d''(x_i) \leq d'(v_i) \quad 1 \leq i \leq n$$

よって

$$d''(x_i) \leq d(\tilde{v}_i) = d(t_i)$$

となり (5.1) が成立する. (5.2), (5.3) が成立することはい明らか.

Case 4 を考える. t_i が変数 x_e を含むのは $q \neq 0$ のとき. このとき u_i は $x_i + w_i$ という term を含む.

したがって Q_1 は $x_i \leftarrow x_i + 1$ という atomic statement を含み, $x_i <_P x_e$ となる. よって (5.3) が成立.

任意の $d \in D$ と $m \in \mathbb{N}$ に対し, $d_m \in D$ を

$$d_m(x_i) = \begin{cases} d(x_i) & \text{if } i \neq l \\ m & \text{if } i = l \end{cases}$$

と定義する。また

$$d'_m = Q_i^m(d)$$

とする。(5.1)を示すには、任意の i , $i \neq l$, と m に対し

$$(5.4) \quad d'_m(x_i) \leq d_m(t_i)$$

が成立することを言えばよい。($i=l$ なら, Q は statement $x_l \leftarrow x_{l+1}$ を含まないから, $Q(d)(x_l) \leq d(x_l) \leq d(t_l)$ が成立する。)

$x_i <_p x_j$ なる任意の j に対し, (5.4) が成立したと仮定する。 m に関する induction で (5.4) を示す。 $m = 0$ なら $d'_m(x_i) = d_m(x_i) \leq d_m(t_i)$ 。よって (5.4) は成立する。 m で成立すると仮定する。任意の term t に対し, $d \leq d'$ なら $d(t) \leq d'(t)$ 。

よって特に

$$d_m(t) \leq d_{m+1}(t)$$

$$d'_{m+1} = Q_i(d'_m) \text{ なり}$$

$$\begin{aligned} d'_{m+1}(x_i) &\leq d'_m(\max\{\max\{v_1, \dots, v_f\}, \\ &\quad x_i + \max\{w_1, \dots, w_g\}, x_i\}) \\ &\leq \max\{\max\{d'_m(v_1), \dots, d'_m(v_f)\}, \\ &\quad d'_m(x_i) + \max\{d'_m(w_1), \dots, d'_m(w_g)\}, \\ &\quad d'_m(x_i)\} \end{aligned}$$

今 $x_i \leq_p x_j$ なる任意の i, j に対し

$$dm(x_j) \leq dm(t_j)$$

が成立する。よって

$$dm(v_x) = dm(\tilde{v}_x), \quad dm(w_x) \leq dm(\tilde{w}_x).$$

帰納法の仮定より

$$dm(x_i) \leq dm(t_i)$$

$$\leq dm(\max\{\max\{\tilde{v}_1, \dots, \tilde{v}_r\} + x_i, \max\{\tilde{w}_1, \dots, \tilde{w}_q\}\})$$

$$\leq dm(x_i + x_e \cdot \max\{\tilde{w}_1, \dots, \tilde{w}_q\})$$

$$\leq \max\{dm(\tilde{v}_1), \dots, dm(\tilde{v}_r), dm(x_i)\}$$

$$+ dm(x_e) \cdot \max\{dm(\tilde{w}_1), \dots, dm(\tilde{w}_q)\}$$

$dm(x_e) = m$ とおくと

$$dm_{m+1}(x_i) \leq \max\{dm(\tilde{v}_1), \dots, dm(\tilde{v}_r), dm(x_i)\}$$

$$+ (m+1) \cdot \max\{dm(\tilde{w}_1), \dots, dm(\tilde{w}_q)\}$$

$$\leq dm_{m+1}(t_i)$$

よって (5.4) が成立する。 [1(2)] の (5.2) を満たすこと

は明らかである。

§ 6 Loop Program on RAM

これまでは基本演算として $+1$, -1 しか許されなかった。ここではこれまでの議論を加減算を許したものに

拡張しよう。

Def. atomic statement とし RAM 上の loop program と呼ぶ。

$$l = f + k \quad | \quad l \leftarrow f - k \quad | \quad x \leftarrow y - z$$

$$x \leftarrow x_1 + x_2 + \dots + x_n$$

ただし、 l, k, f, k は variable, x, y, z は control variable $x_1, \dots, x_n (n \geq 1)$ は control variable または constant である。 $x \leftarrow x_1 + \dots + x_n$ の形の statement を key statement と言う。P に現われる key statement の個数を P の width といい $\text{width}(P)$ で示す。

本節では、単に loop program とさえし、RAM 上の loop program のことを言う。

Def. P を loop program とする。このとき $x \neq y$ が成立するのとは次の場合である。

- (i) P が $\text{loop } x \text{ do } Q \text{ end}$ の形の statement を含み、Q が $y \leftarrow \alpha$ の形の key statement を含む。
- (ii) x と y が異なる変数で、P が $y \leftarrow \alpha$ の形の key statement を含み、 α が x を含む。

loop program P が simple であるとは $x_1 \neq x_2 \neq \dots \neq x_n,$
 $n \geq 1$

なる列が存在しないときをいう。

Def. $q(k, n)$ を次で定義する。

$$q(0, n) = n$$

$$q(k+1, n) = 2^{q(k, n)}$$

即ち $q(k, n)$ は n の k 重の指数関数である。各 k に対し、 Σ_{xp}^k である多項式 P が存在し、 $q(k, p(n))$ 時間で認識できる言語のクラスを表わす、(Σ_{xp}^k は TM, RAM, SM のいずれの模型上で考えても同じクラスが定義されることに注意する。) したがって $\mathcal{P} = \Sigma_{xp}^0$ であり、 $\cup \Sigma_{xp}^k$ は初等的問題全体からなるクラスである。

以下で、クラス Σ_{xp}^k が width によって特長づけられることを示す。特に言語 L が初等的である必要十分条件は L が simple loop program on RAM で accept されることである。証明の方針は §3, §4 と同様である。

Def. reduced form, normal form を次のように定義する。

ここで x は control variable, $x_1, \dots, x_n, n \geq 0$ は control variable, s は constant, l は variable,

$$\langle \text{reduced form} \rangle ::= x \leftarrow x + x_1 + \dots + x_n \mid l \leftarrow 0 \mid$$

$$\langle \text{reduced form} \rangle ::= \langle \text{reduced form} \rangle \mid$$

loop x do $\langle \text{reduced form} \rangle$ end

$\langle \text{type 1} \rangle ::= \langle \text{key statement} \rangle \mid \text{loop } x \text{ do } \langle \text{type 1} \rangle \text{ end}$

$\langle \text{normal form} \rangle ::= \langle \text{type 1} \rangle \mid \langle \text{normal form} \rangle ; \langle \text{normal form} \rangle$

Lemma 6.1 任意の loop program P on RAM に対し,
reduced \bar{P} の存在し次を満たす.

(i) $\text{width}(P) = \text{width } \bar{P}$

(ii) $\text{time}_P(d) \leq \text{time}_{\bar{P}}(d)$ for all $d \in D$

Proof. P の各 substatement Q に対し, Q の reduced form \bar{Q} を次のように定める. x は P に現われる control variable とする. Q の key statement $x \leftarrow x + 1$ ならば \bar{Q} は $x \leftarrow x + 1$, Q の key statement 以外の atomic statement ならば \bar{Q} は $x \leftarrow 0$. Q が Q_1, Q_2 の $\text{if } g \text{ then } Q_1 \text{ else } Q_2$ ならば \bar{Q} は \bar{Q}_1, \bar{Q}_2 . Q が $\text{loop } x \text{ do } Q_1 \text{ end}$ ならば \bar{Q} は $\text{loop } x \text{ do } \bar{Q}_1, \text{end}$. すると \bar{P} は lemma を満たす.

Theorem 3.6 と同様にして次の示される.

Theorem 6.2 P は reduced かつ simple loop program とする. このとき simple loop program Q of normal form

て次を満たすものが存在する。

$$(i) \quad \text{width}(P) = \text{width}(Q)$$

$$(ii) \quad \text{time}_P(d) \leq \text{time}_Q(d) \quad \text{for all } d \in D$$

Theorem 6.3 P is simple loop program on RAM,

$k = \text{width}(P)$ とする。 $d \in D$, $n = \max_{x \in X} d(x)$ とする。

すると多項式 $P(n)$ が存在し

$$\text{time}_P(d) \leq g(k, P(n))$$

Proof. Lemma 6.1 と Theorem 6.2 より, P is simple loop program of normal form としてあり

$$P = P_1; P_2; \dots; P_n$$

とする。 $i=1$ から各 P_i は type 1, m に関する帰納法で, polynomial $p(n)$ が \exists する

$$(6.1) \quad \text{time}_P(d) \leq g(k, p(n))$$

$$(6.2) \quad P(d)(x) \leq g(k, p(n)) \quad \text{for all } x \in X.$$

が成立すると仮定する。 m に対して成立し i とし

$$P' = P, P_{m+1}$$

を考える。 $\text{width}(P_{m+1}) = 0$ なら $\text{width}(P') = k$.

P_{m+1} は control variable を含みないので

$$P_{m+1}(d)(x) = d(x) \quad \text{for all } x \in X.$$

よって (6.2) より

$$P'(d)(x) \leq g(k, p(n))$$

が成立する。また P_{m+1} のネストの深さが ℓ より

$$\text{time}_{P_{m+1}}(d) \leq (\max_x d(x))^{\ell}$$

よって

$$\begin{aligned} \text{time}_{P'}(d) &\leq g(k, p(n)) + (g(k, p(n)))^{\ell} \\ &\leq g(k, p(n)^{\ell+1} + \ell + 1) \end{aligned}$$

よって (6.1) が成立。 P_{m+1} かつ

$$\text{loop } x_1, \text{ loop } x_2, \dots, \text{ loop } x_c \text{ do } x \leftarrow y_1 + y_2 + \dots + y_c \text{ end} \dots \text{end end}$$

の形のとき $n = \max_{y \in x} d(y)$, $\lambda \leq 2^c$, $c \geq 1$, とすると

$$P_{m+1}(d)(x) \leq \lambda^{ne} \leq 2^{cne}$$

$$\text{time}_{P_{m+1}}(d) \leq \lambda^{ne} \leq 2^{cne}$$

よって

$$\begin{aligned} \text{time}_{P'}(d) &\leq g(k, p(n)) + 2^{cg(k, p(n))e} \\ &\leq g(k+1, (c+1)(p(n)^{\ell+1} + \ell + c + 1)) \end{aligned}$$

よって (6.1), (6.2) が成立する。

Lemma 4.1 と同様に次を示される。

Lemma 6.4 $L \in \mathcal{E}_{xp}^k$ なら, L は $\text{width}(P) = k$ の simple loop program τ で受理される。

Proof P_i, R を次のように定める。

$$P_i = \text{loc } x_i \text{ do } x_{i+1} \leftarrow x_i + x_i \text{ end}$$

$$R = P_1; P_2, \dots, P_k$$

とすると $d(x) = n$ とすると

$$R(d)(x_{k+1}) \geq f(k, n)$$

となる。あとは Lemma 4.1 と同じ。

Corollary 6.5 $L \in E_{xp}^k$ である必要十分条件は width が k の simple loop program on RAM のプログラムで L が受理されること。

§ 7 Conclusion

ここでは制限されたプログラムのクラスについて、その時間量の上限が syntactical に解析できることを示した。同様の方法で space complexity の上限も計算できる。[1]。また配列の添字変数がその配列の範囲を越えないことのチェックもできるであらう。たとえば Algorithm 5.1 で得られる $V(P) = (t_1, \dots, t_n)$ を使うと memory map d のもとでの P の計算の任意の時点の memory d' に対し

$$d'(x_i) \leq d(t_i)$$

が成立する。いいかえれば変数 x_i の内容は計算の途中

で値 $\alpha(x_i)$ を越えない。

ここで得られた理論を現実の programming language に適用するにはさらにいくつかの考察が必要であらう。たとえば while q do α end といった while statement は loop statement と組み合わせ

loop t while q do α end

といったように使うように制限を付けなければならぬ。

ここで t は このループをまわる回数の上限を表わす term である。 syntax をゆるめると同時に, Simpleness の条件も拡張する必要がある。たとえば § 6 でみないように RAM 上の loop program では simpleness の条件だけでは不十分である。

文献

- [1] Adachi, A., A relation between space complexity and the largest number (未刊論文)
- [2] Borodin, A., Computational complexity: theory and practice, in: Currents in the Theory of Computing, A.V. Aho, ed., Prentice-Hall, Englewood Cliffs, N.J., 1973. 35-89. 宇屋悦朗訳「最近の計算機理論」近代科学社
- [3] Constable, R.L., On the size of programs in subrecursive hierarchies. Proc. 2nd Annual ACM Symposium on Theory of Computing (1970) 1-9
- [4] Constable, R.L., and A. Borodin, On the efficiency of programs in subrecursive formalisms. IEEE Conference Record 11th Annual Symposium on Switching and Automata Theory (1970) 60-67
- [5] Cook, S.A., and R.A. Reckhow, Time bounded random access machines, J. Comput. Syst. Sci. 7 (1973), 354-375
- [6] Jones, N.D., and S.S. Muchnick, Even simple programs are hard to analyze, J. ACM 24 (1977) 338-350

- [7] Kasai, T., Computational complexity of multitape Turing machines and random access machines, Publ. RIMS, Kyoto Univ. 13 (1977), 469-496
- [8] Kasai, T., $P=NP$ question on program schemata, Proc. Second IBM Symp. on Mathematical Foundations of Computer Science, IBM Japan, 1977
- [9] Meyer, A.R., and D.M. Ritchie, The complexity of loop programs. Proc. 22nd National ACM Conference, (1967) 465-470
- [10] Grzegorzczuk, A., Some class of recursive functions, Rozprawy Matematyczne, No. 4 Instytut Matematyczny Polskiej Akademii Nauk, Warsaw, Poland 1953 pp. 1-45